SQL: The Complete Reference

James R. Groff
Paul N. Weinberg

Osborne/McGraw-Hill
Berkeley New York St. Louis San Francisco
Auckland Bogotá Hamburg London Madrid
Mexico City Milan Montreal New Delhi Panama City
Paris São Paulo Singapore Sydney
Tokyo Toronto

Osborne/McGraw-Hill
2600 Tenth Street
Berkeley, California 94710
U.S.A.

For information on translations or book distributors outside the U.S.A., or to arrange bulk purchase discounts for sales promotions, premiums, or fund-raisers, please contact Osborne/McGraw-Hill at the above address.

SQL: The Complete Reference

1234567890 DOC DOC 90198765432109

ISBN 0-07-211845-8

Publisher
Brandon A. Nordin

Associate Publisher and Editor-in-Chief
Scott Rogers

Senior Acquisitions Editor
Wendy Rinaldi

Acquisitions Editor
Jane K. Brownlow

Project Editor
Heidi Poulin

Editorial Assistant
Monika Faltiss

Copy Editor
Nancy Crumpton

Proofreader
Rhonda Holmes

Indexer
Valerie Robbins

Computer Designer
Jani Beckwith
Michelle Galicia

Illustrators
Robert Hansen
Brian Wells
Beth Young

In many ways, queries are the heart of the SQL language. The SELECT statement, which is used to express SQL queries, is the most powerful and complex of the SQL statements. Despite the many options afforded by the SELECT statement, it's possible to start simply and then work up to more complex queries. This chapter discusses the simplest SQL queries—those that retrieve data from a single table in the database.

The SELECT Statement

The SELECT statement retrieves data from a database and returns it to you in the form of query results. You have already seen many examples of the SELECT statement in the quick tour presented in Chapter 2. Here are several more sample queries that retrieve information about sales offices:

List the sales offices with their targets and actual sales.

```
SELECT CITY, TARGET, SALES
  FROM OFFICES'

CITY                TARGET         SALES
------------  ------------  ------------
Denver         $300,000.00   $186,042.00
New York       $575,000.00   $692,637.00
Chicago        $800,000.00   $735,042.00
Atlanta        $350,000.00   $367,911.00
Los Angeles    $725,000.00   $835,915.00
```

List the Eastern region sales offices with their targets and sales.

```
SELECT CITY, TARGET, SALES
  FROM OFFICES
 WHERE REGION = 'Eastern'

CITY                TARGET         SALES
------------  ------------  ------------
New York       $575,000.00   $692,637.00
Chicago        $800,000.00   $735,042.00
Atlanta        $350,000.00   $367,911.00
```

List Eastern region sales offices whose sales exceed their targets, sorted in alphabetical order by city.

```
SELECT CITY, TARGET, SALES
  FROM OFFICES
 WHERE REGION = 'Eastern'
   AND SALES > TARGET
 ORDER BY CITY

CITY                TARGET         SALES
------------  ------------  ------------
Atlanta        $350,000.00   $367,911.00
New York       $575,000.00   $692,637.00
```

What are the average target and sales for Eastern region offices?

```
SELECT AVG(TARGET), AVG(SALES)
  FROM OFFICES
 WHERE REGION = 'Eastern'

 AVG(TARGET)    AVG(SALES)
------------  ------------
 $575,000.00   $598,530.00
```

For simple queries, the English language request and the SQL SELECT statement are very similar. When the requests become more complex, more features of the SELECT statement must be used to specify the query precisely.

Figure 6-1 shows the full form of the SELECT statement, which consists of six clauses. The SELECT and FROM clauses of the statement are required. The remaining four clauses are optional. You include them in a SELECT statement only when you want to use the functions they provide. The following list summarizes the function of each clause:

- The SELECT clause lists the data items to be retrieved by the SELECT statement. The items may be columns from the database, or columns to be calculated by SQL as it performs the query. The SELECT clause is described in later sections of this chapter.
- The FROM clause lists the tables that contain the data to be retrieved by the query. Queries that draw their data from a single table are described in this chapter. More complex queries that combine data from two or more tables are discussed in Chapter 7.
- The WHERE clause tells SQL to include only certain rows of data in the query results. A *search condition* is used to specify the desired rows. The basic uses of the WHERE clause are described later in this chapter. Those that involve subqueries are discussed in Chapter 9.

- The GROUP BY clause specifies a summary query. Instead of producing one row of query results for each row of data in the database, a summary query groups together similar rows and then produces one summary row of query results for each group. Summary queries are described in Chapter 8.
- The HAVING clause tells SQL to include only certain groups produced by the GROUP BY clause in the query results. Like the WHERE clause, it uses a search condition to specify the desired groups. The HAVING clause is described in Chapter 8.
- The ORDER BY clause sorts the query results based on the data in one or more columns. If it is omitted, the query results are not sorted. The ORDER BY clause is described later in this chapter.

The SELECT Clause

The SELECT clause that begins each SELECT statement specifies the data items to be retrieved by the query. The items are usually specified by a *select list*, a list of *select items* separated by commas. Each select item in the list generates a single column of query results, in left-to-right order. A select item can be:

- a *column name*, identifying a column from the table(s) named in the FROM clause. When a column name appears as a select item, SQL simply takes the value of that column from each row of the database table and places it in the corresponding row of query results.
- a *constant*, specifying that the same constant value is to appear in every row of the query results.
- a *SQL expression*, indicating that SQL must calculate the value to be placed into the query results, in the style specified by the expression.

Each type of select item is described later in this chapter.

The FROM Clause

The FROM clause consists of the keyword FROM, followed by a list of table specifications separated by commas. Each table specification identifies a table containing data to be retrieved by the query. These tables are called the *source tables* of the query (and of the SELECT statement) because they are the source of all of the data in the query results. All of the queries in this chapter have a single source table, and every FROM clause contains a single table name.

Query Results

The result of a SQL query is always a table of data, just like the tables in the database. If you type a SELECT statement using interactive SQL, the DBMS displays the query

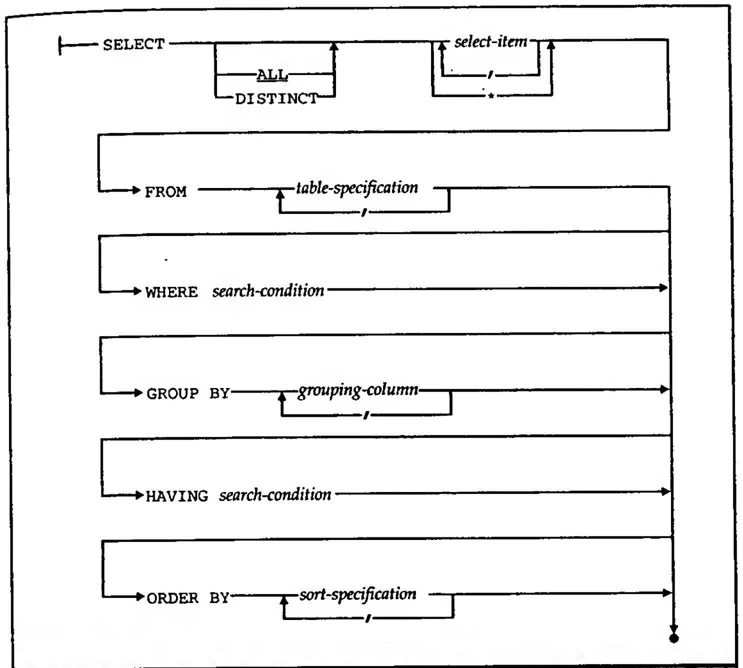

Figure 6-1. SELECT statement syntax diagram

results in tabular form on your computer screen. If a program sends a query to the DBMS using programmatic SQL, the table of query results is returned to the program. In either case, the query results always have the same tabular, row/column format as the actual tables in the database, as shown in Figure 6-2. Usually the query results will be a table with several columns and several rows. For example, this query produces a table of three columns (because it asks for three items of data) and ten rows (because there are ten salespeople):